

Using Configurable CPUs on SOC Platforms

5th Int'l Forum on Application-Specific Multi-Processor SoC
11 - 15 July 2005, Relais de Margaux, France



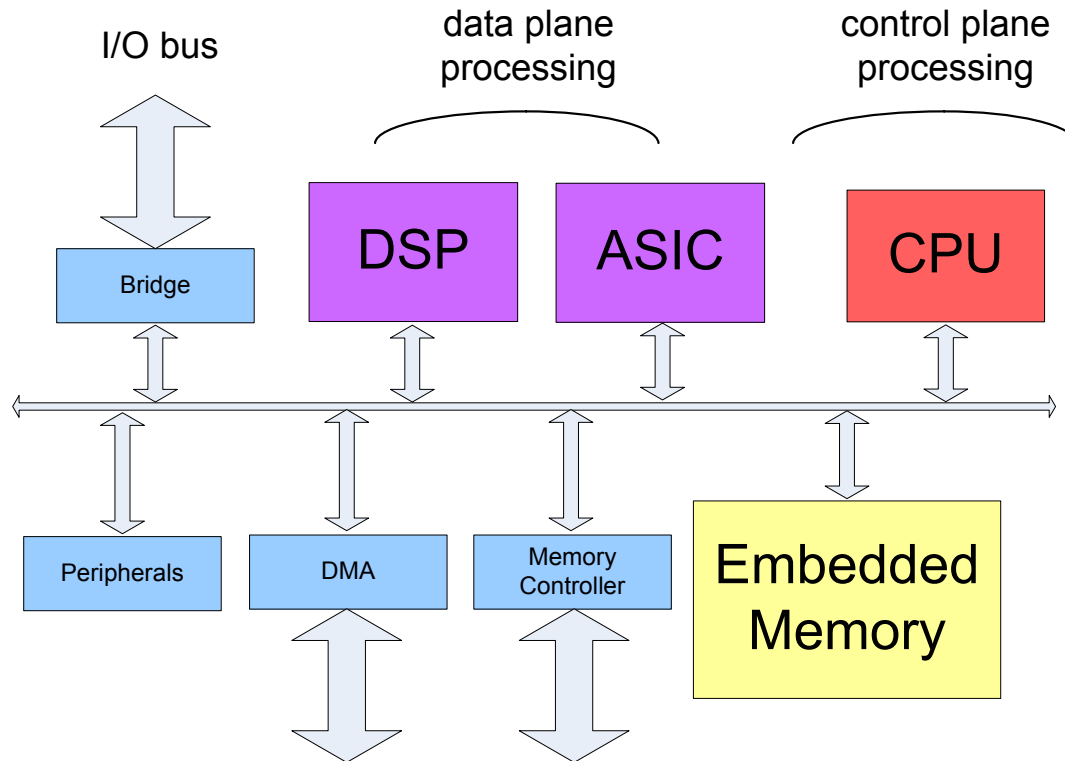
Trevor Mudge

Bredt Family Professor of Engineering

The University of Michigan, Ann Arbor

Typical SoC

- Custom configurations of existing (IP) cores – OMAP
 - control plane processors – ARM/MIPS/etc.
 - data plane processor – TI 5000/6000/etc.
 - ASICs + glue logic



Why Use Configurable CPUs

1. Reduce power and meet performance goals:
 - By absorbing data plane functionality into the control plane CPU
 - tightly coupled – new instructions created
 - design time – Tensilica, ARC
 - By augmenting or replacing data plane DSP
 - loosely coupled co-processor – separate instruction stream
 - design time – ARM+Optimode
 - run-time – Xilinx, Altera
 - FPGA solution+embedded processor
2. As a flexible alternative to ASICs
 - Programmability “future proofs” the SOC
 - SOC decreases time-to-market thru reuse
 - SOC : 1 year design time
 - ASIC : 2.5 year design time
 - Using flexible-configurable logic enables further reduction in time-to-market

Candidates for Configurable CPUs

Compute intensive inner loops or kernels

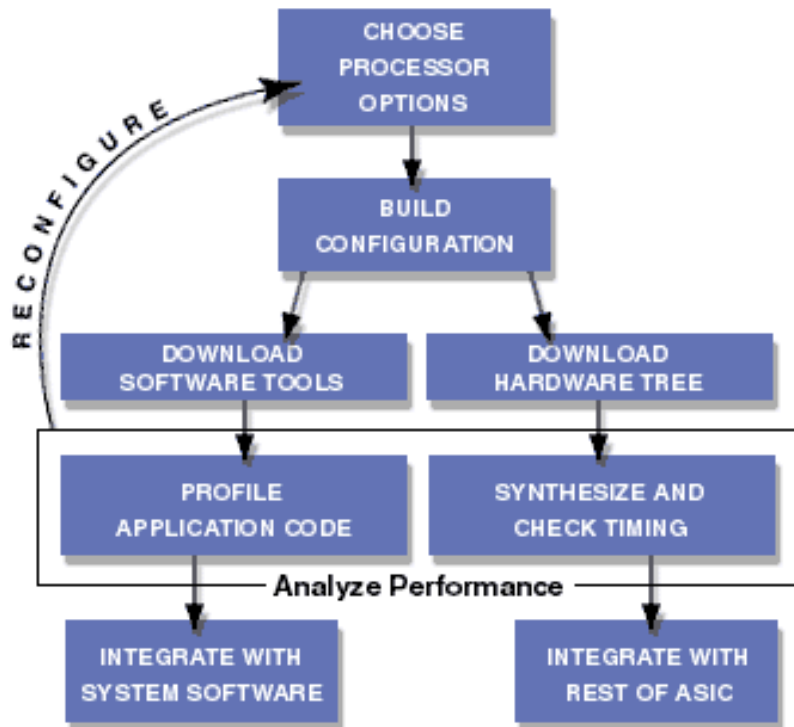
- inner loops identified and new instructions or co-processor tailored to their efficient execution
- Examples of instructions:
 - a MAC instruction
 - `for (i = 0; i < n; i++) sum = sum + a[i] * b[i];`
 - add-compare-select
- Examples of co-processors
 - Fast Fourier Transform
 - Viterbi decoder

Tightly Coupled and Configurable at Design Time

Examples: Tensilica, ARC

- Customization added to a basic microprocessor base
- Typically the fetch unit, ALU, load/store unit and basic control units in the processor are fixed
- Configurable options
 - register file, additional execution Units – except basic ALU
 - cache/memory type, size
 - debug support
 - interrupts, timers, peripherals
- New instructions/functionality added during design space exploration
- RTL code, compiler, instruction level full system simulation and the description of the architecture is generated
 - simulation studies allow performance/power calibration
 - EDA tools are later used to implement these configurable processors

Design Flow for Configurable Processors



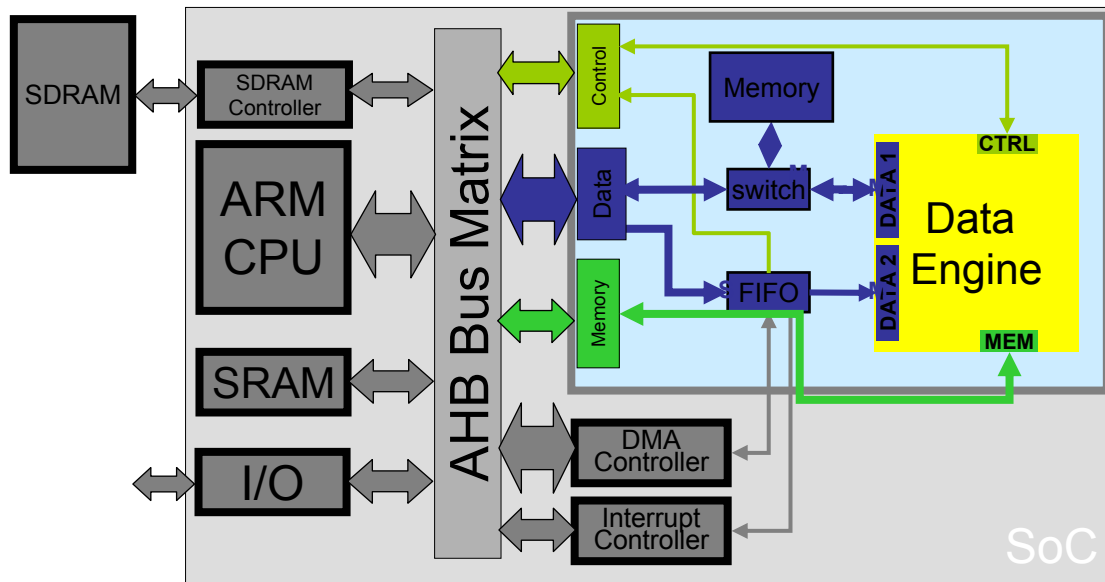
- The customizable processor can be optimized for the application in terms of
 - gate count
 - power consumption
 - performance

Tensilica figures compared to a general purpose 32-bit RISC Core (ARM or MIPS):
7x improvement on W-CDMA applications
50x improvement on MPEG4 QCIF
250x improvement on GSM Viterbi Decoding

Loosely Coupled and Configurable at Design Time

Example: Optimode+ARM

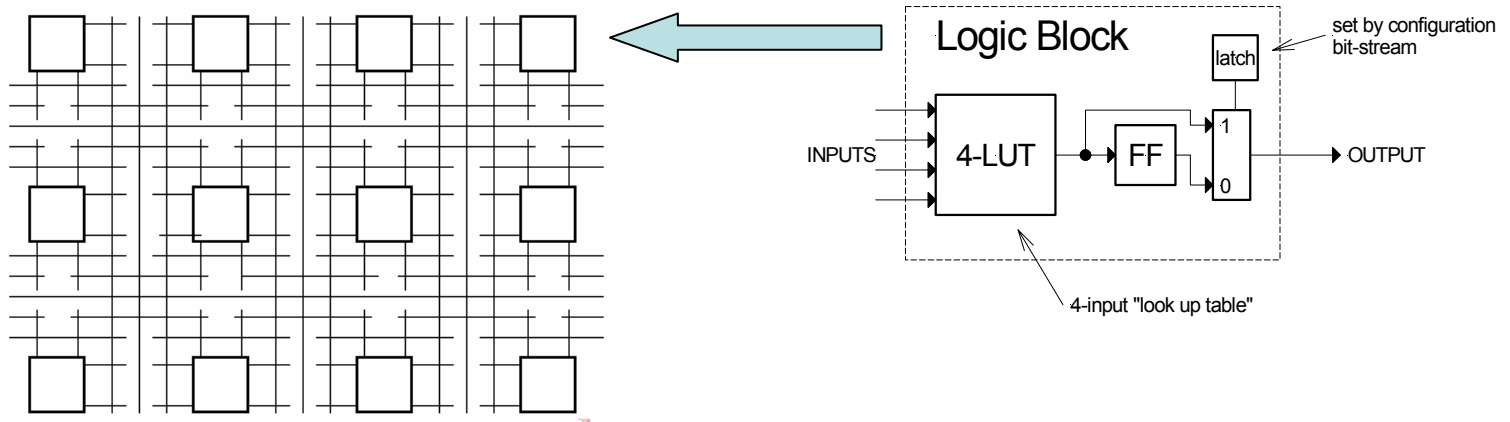
- Separate co-processor with own instruction stream
 - horizontally microcoded engine
- Design flow is geared to iteratively improve kernel performance
 - C-like description



Configurable at Run Time

Examples – FPGAs: Xilinx, Altera

- 2D array of logic blocks and flip-flops with configurable
 - interconnect between the logic blocks & their function
- Configuration stored in RAM
- Embedded hard cores
 - Multipliers
 - Processors: Xilinx 32-bit PowerPC; Altera 32-bit ARM
- Softcores
 - Processors: Xilinx 8-bit PicoBlaze, 32-bit MicroBlaze; Altera 32-bit Nios



- Development cycle is less integrated into an SOC flow
- Power a consideration